...e costruisci il tuo LABORATORIO DIGITALE

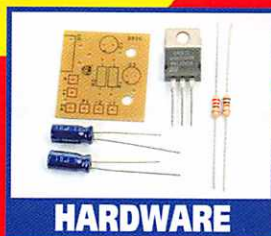
6,90 €

61

HARDWARE

DIGITALE DI BASE

MICROCONTROLLER

DIGITALE AVANZATO

TOTALMENTE PROGRAMMABILE!!!

9 771824 363008
41061

Peruzzo & C.

impara l'elettronica digitale

IN REGALO in questo fascicolo

- 1 Scheda DG20
- 1 Circuito integrato LM317T
- 1 Resistenza da 220 Ω 5% 1/4 W
- 1 Resistenza da 10 K 5% 1/4 W
- 2 Condensatori elettrolitici da 10 µF

IN REGALO nel prossimo fascicolo

- 1 Potenziometro con alberino metallico da 2K5 con bullone e rondella
- 1 Manopola per potenziometro
- 3 Viti

COME RACCOGLIERE E SUDDIVIDERE L'OPERA NELLE 4 SEZIONI

L'Opera è composta da 4 sezioni identificabili dalle fasce colorate, come indicato sotto. Le schede di ciascun fascicolo andranno suddivise nelle sezioni indicate e raccolte nell'apposito raccoglitore, che troverai presto in edicola. Per il momento, ti consigliamo di suddividere le sezioni in altrettante cartellette, in attesa di poterle collocare nel raccoglitore. A prima vista, alcuni numeri di pagina ti potranno sembrare ripetuti o sbagliati. Non è così: ciascuno fa parte di sezioni differenti e rispecchia l'ordine secondo cui raccogliere le schede. **Per eventuali domande di tipo tecnico scrivere al seguente indirizzo e-mail:** elettronicadigitale@microrobots.it

Hardware Montaggio e prove del laboratorio

Digitale di base Esercizi con i circuiti digitali

Digitale avanzato Esercizi con i circuiti sequenziali

Microcontroller Esercizi con i microcontroller

Direttore responsabile:
ALBERTO PERUZZO
Direttore Grandi Opere:
GIORGIO VERCELLINI
Consulenza tecnica e traduzioni:
CONSULCOMP S.n.c.
Pianificazione tecnica
LEONARDO PITTON

Direzione, Redazione, Amministrazione: viale Ercole Marelli 165, Tel. 02/242021, 20099 Sesto San Giovanni (Mi). Pubblicazione settimanale. Registrazione del Tribunale di Monza n. 1738 del 26/05/2004. Spedizione in abbonamento postale gr. II/70; autorizzazione delle Poste di Milano n. 163464 del 13/2/1963. Stampa: Grafiche Porpora s.r.l., Cernusco S/N (MI). Distribuzione SO.DI.P. S.p.A., Cinisello Balsamo (MI).

"ELETTRONICA DIGITALE" si compone di 70 fascicoli settimanali da suddividere in 2 raccoglitori.

RICHIESTA DI NUMERI ARRETRATI. Per ulteriori informazioni, telefonare dal lunedì al venerdì ore 9.30-12.30 all'ufficio arretrati tel. 02/242021. Se vi mancano dei fascicoli o dei raccoglitori per completare l'opera, e non li trovate presso il vostro edicolante, potrete riceverli a domicilio rivolgendovi direttamente alla casa editrice. Basterà compilare e spedire un bollettino di conto corrente postale a PERUZZO & C. s.r.l., Ufficio Arretrati, viale Marelli 165, 20099 Sesto San Giovanni (MI). Il nostro numero di c/c postale è 42980201. L'importo da versare sarà pari al prezzo dei fascicoli o dei raccoglitori richiesti, più le spese di spedizione € 3,10 per pacco. Qualora il numero dei fascicoli o dei raccoglitori sia tale da superare il prezzo globale di € 25,82 e non superiore a € 51,65, l'invio avverrà per pacco assicurato e le spese di spedizione ammonteranno a € 6,20. La spesa sarà di € 9,81 da € 51,65 a € 103,29; di € 12,39 da € 103,29 a € 154,94; di € 14,98 da € 154,94 a € 206,58; di € 16,53 da € 206,58 in su. Attenzione: ai fascicoli arretrati, trascorse dodici settimane dalla loro distribuzione in edicola, viene applicato un sovrapprezzo di € 0,52, che andrà pertanto aggiunto all'importo da pagare. Non vengono effettuate spedizioni contrassegno. Gli arretrati di fascicoli e raccoglitori saranno disponibili per un anno dal completamento dell'opera. IMPORTANTE: è assolutamente necessario specificare sul bollettino di c/c postale, nello spazio riservato alla causale del versamento, il titolo dell'opera nonché il numero dei fascicoli e dei raccoglitori che volete ricevere.

Alimentazione variabile

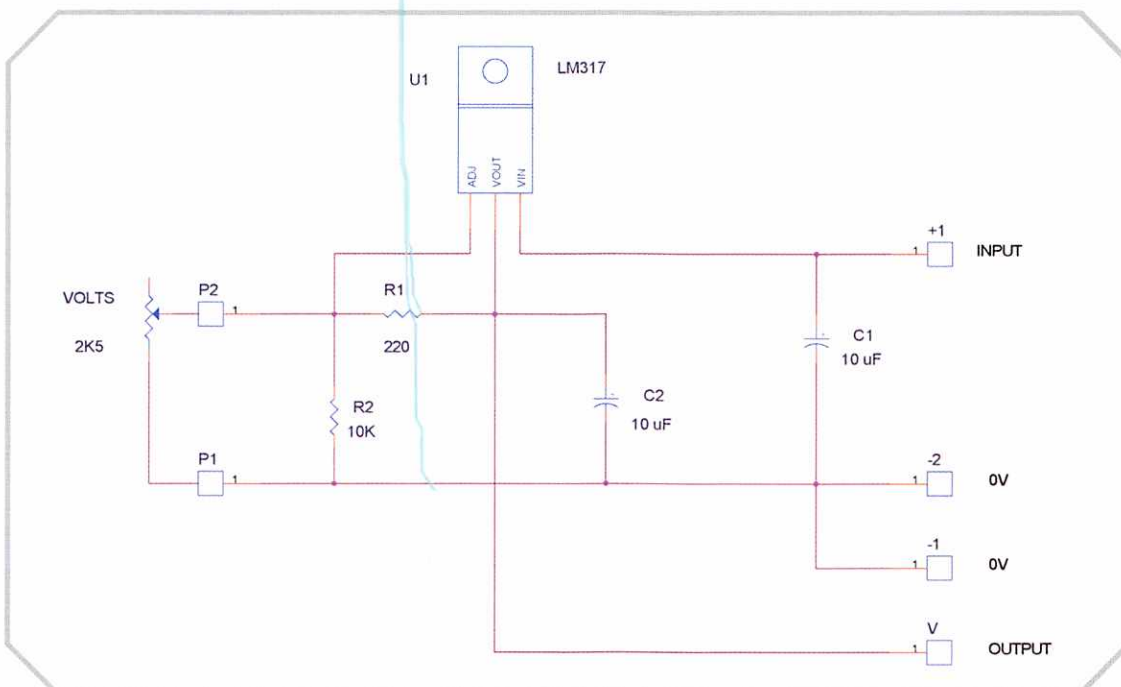
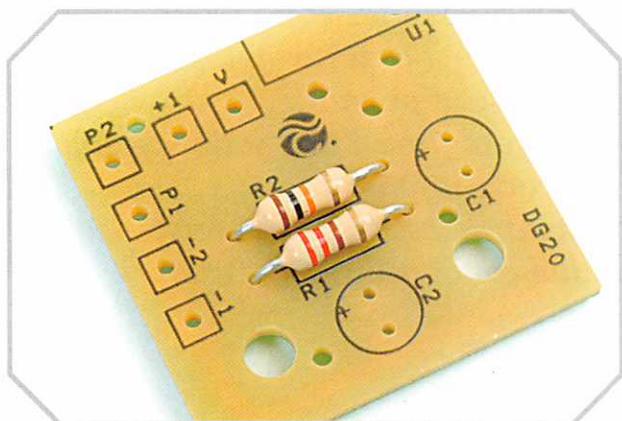
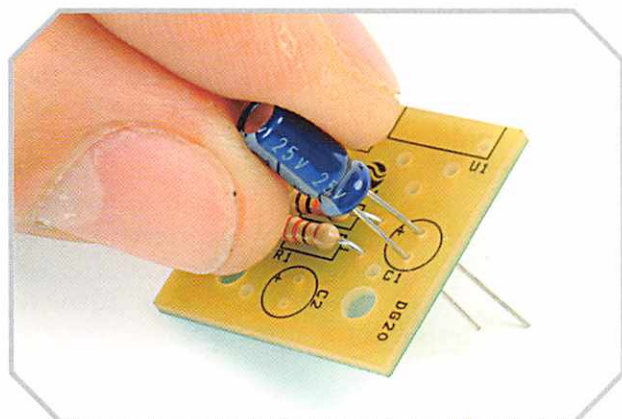
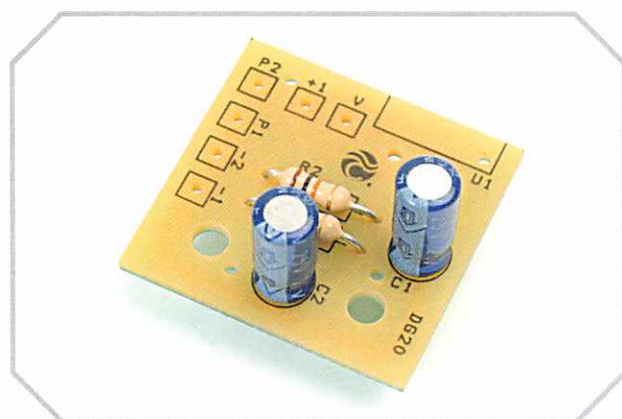
Con questo fascicolo viene fornito il circuito stampato DG20 e tutti i componenti che andranno saldati sullo stesso: il circuito integrato regolatore LM317, due resistenze e due condensatori elettrolitici.

Materiale allegato a questo fascicolo.

Circuito stampato DG20.

Il circuito

Se osserviamo lo schema, che appare molto semplice per il ridotto numero di componenti, possiamo vedere che si tratta di un circuito per regolare la tensione, e la sua semplicità è dovuta all'impiego del circuito integrato LM317, componente molto utilizzato nella costruzione di alimentatori regolabili.

Uno dei grandi vantaggi nell'utilizzo di questo tipo di regolatori integrati è che evitano l'impiego di elementi di regolazione. Un altro vantaggio è che necessitano di pochissimi componenti esterni addizionali.

Schema elettrico dello stabilizzatore regolabile tra 2 e 12 volt.

Le due resistenze montate.

Fate attenzione alla polarità dei condensatori.

Condensatori montati.

La tensione di uscita di questo circuito si può regolare con il potenziometro VOLTS, che sarà fornito prossimamente, tra 2 e 12 volt, ammesso che il laboratorio sia alimentato con una tensione continua da 14 volt. L'uscita di questo circuito, una volta installato, sarà disponibile sulle molle di collegamento V, situate vicino alla scheda Bread Board.

Montaggio del DG20

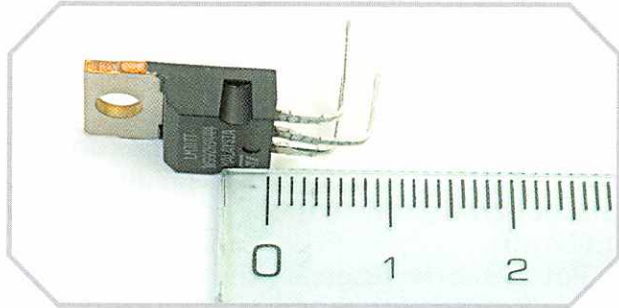
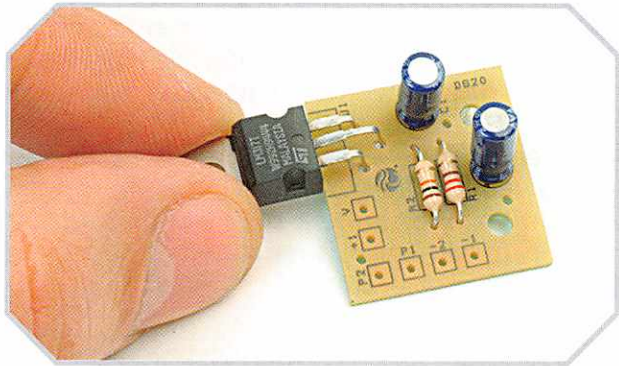
Il montaggio dei componenti sulla scheda inizia come d'abitudine dalle resistenze R1 da 220 Ω (rosso, rosso, marrone), e R2 da 10 K (marrone, nero, arancio). Dopo aver inserito i terminali, li salderemo e poi taglieremo la parte in eccesso dei reofori.

I condensatori

Continuiamo il montaggio con l'inserimento dei due condensatori elettrolitici, entrambi da 10 µF, rispettandone la polarità: il terminale positivo, normalmente più lungo, si inserisce nel foro siglato con il segno + sul circuito stampato. La polarità è indicata anche sul contenitore del componente, normalmente con una striscia colorata in grigio o in nero, che deve fare da riferimento in ogni caso poiché, anche se il condensatore esce dalla fabbrica con il terminale positivo più lungo, è possibile che in qualche punto della catena di distribuzione possa essere stato manipolato e i suoi terminali tagliati a misura diversa.

Terminali dell'integrato

Per poter installare il circuito integrato in modo corretto è necessario piegare prima i suoi terminali, facilitando così l'inserzione degli stessi sul circuito stampato. Questo lavoro si può realizzare facilmente utilizzando delle pinzette a punta piatta, e seguendo le fotografie dove è mostrata la forma che devono assumere questi terminali. Verificheremo anche il circuito stampato per valutare la posizione dei tre fori.

Prima di piegare i terminali bisogna essere sicuri di ciò che stiamo facendo, perché se dovessimo piegare più volte i terminali per correggere la posizione, rischieremmo di romperli.

Circuito integrato con i suoi terminali piegati.

Inserzione del circuito integrato.

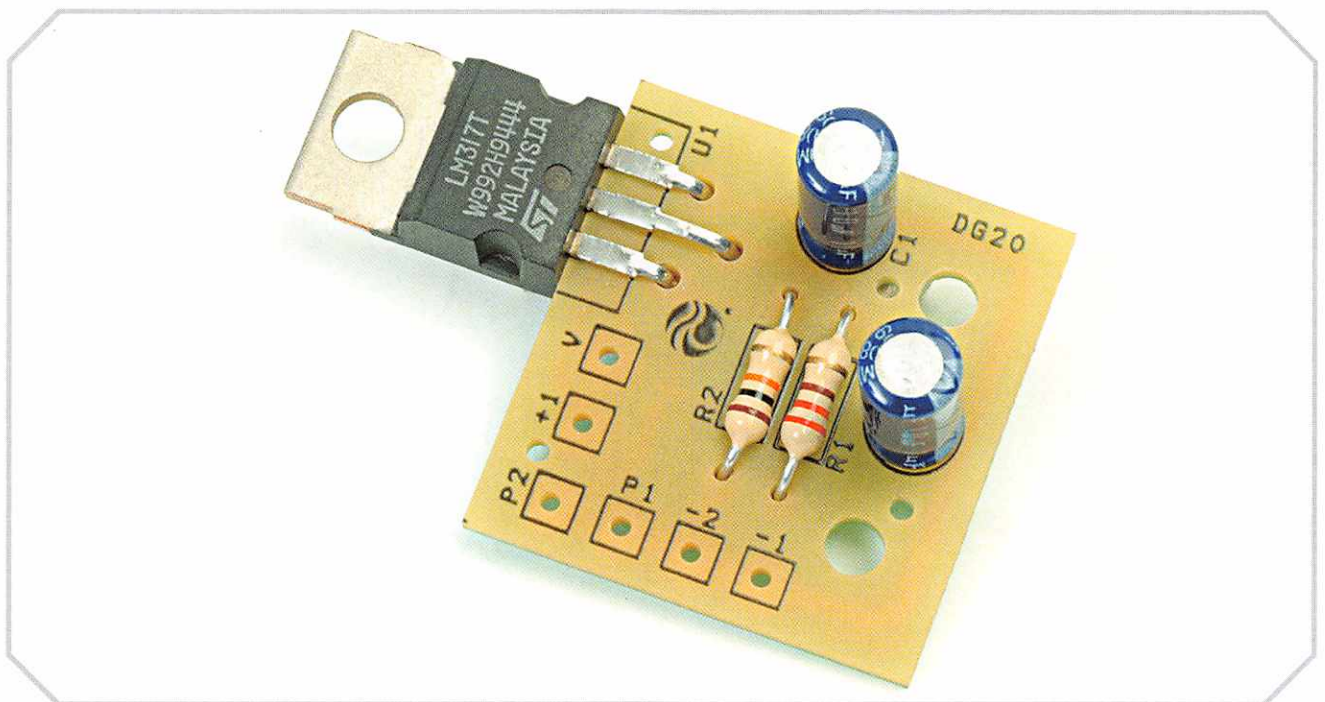
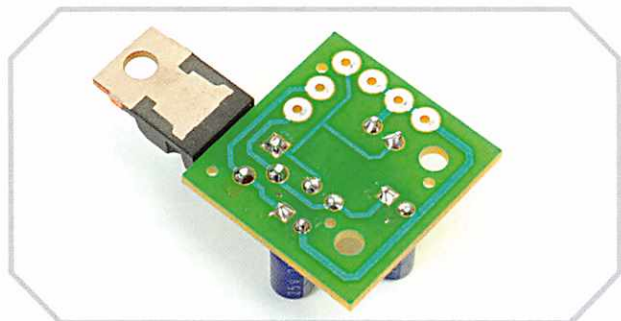
Installazione del circuito integrato

I terminali del circuito integrato si inseriscono sul circuito stampato dal lato dei componenti ed è necessario verificare che tutti e tre fuoriescano per una misura sufficiente dall'altro lato per facilitarne la saldatura. È necessario collocare il circuito integrato nella sua posizione definitiva prima di saldarne i terminali. La saldatura deve essere eseguita con attenzione applicando la punta del saldatore il tempo sufficiente per ottenere una saldatura di buona qualità.

Posizione

Questa scheda trova posto all'interno del pannello superiore del laboratorio, sopra il potenziometro VOLTS, si fissa grazie ai due fori ed è progettata perché possa fornire un massimo di 100 mA senza dissipatore.

Installazione

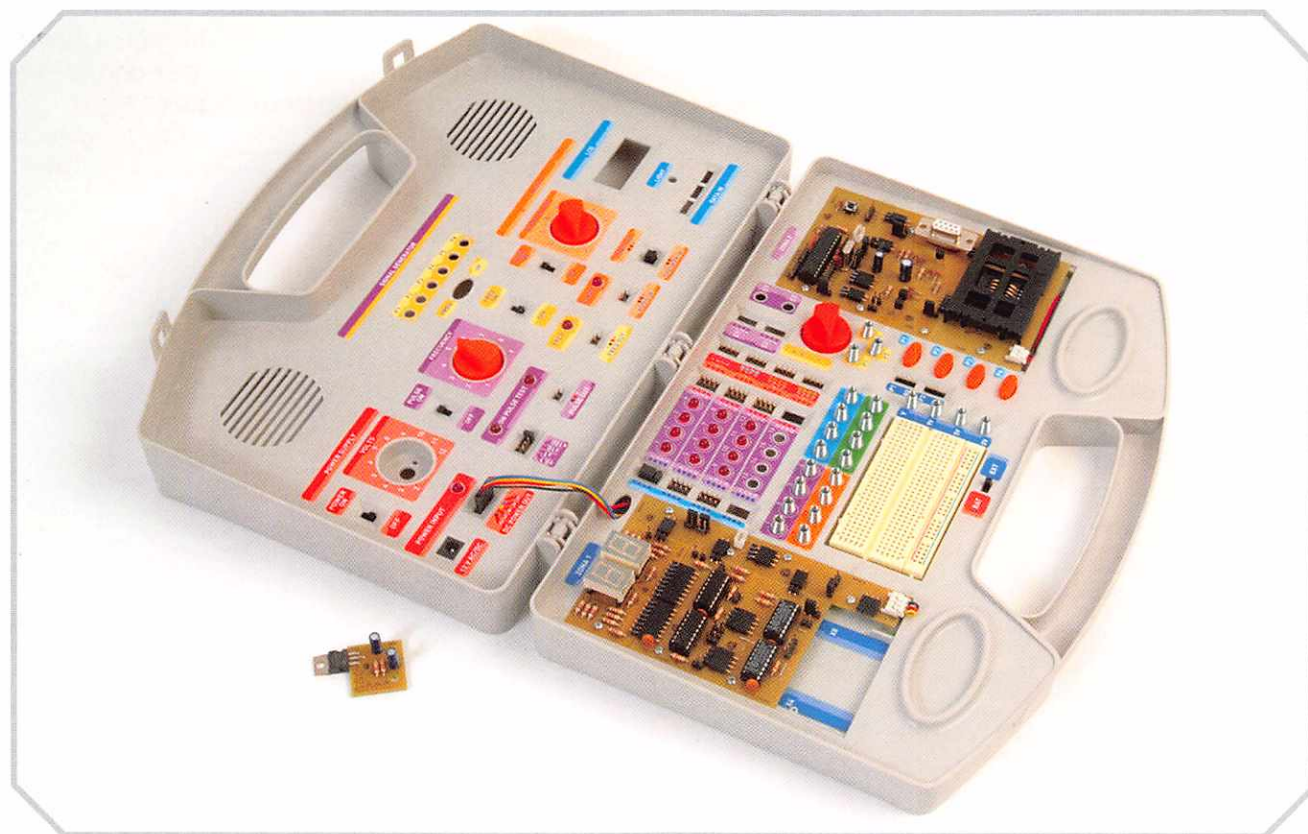
Occorre attendere il prossimo numero per eseguire l'installazione, dato che verrà fornito il potenziometro, necessario per completare il regolatore e poterlo utilizzare.

DG20 con l'integrato saldato.

Dettaglio delle saldature.

La scheda si monterà in questa zona.

Progetto aperto

Questo circuito è stato progettato pensando al fatto che gli utenti più esperti come studenti di elettronica, possano modificarlo con facilità, per esempio aggiungendo un dissipatore per utilizzarlo con tensioni maggiori, fino a 1.000 mA.

Potrete anche osservare che a questo scopo esiste la possibilità di utilizzare grandi radiatori avvitati, dato che si possono aggiungere più viti, ma questo materiale è necessario solo per i più esperti che continueranno a utilizzare il laboratorio adattandolo alle loro esigenze; tutto questo supera le finalità di questa opera quindi questo materiale non viene fornito, dato che non è necessario per gli esperimenti proposti.

Osservando la posizione della scheda possiamo vedere che si trova vicino a una griglia di ventilazione simile a quella utilizzata per fare in modo che il suono dell'altoparlante fuoriesca verso l'esterno.

Laboratorio prima dell'installazione della scheda DG20.

Temporizzatore con controllo acustico

Collegando il circuito si genera in modo continuo un segnale acustico, il cui suono si modifica in modo netto quando viene attivato.

Il circuito

In alcuni casi può essere molto utile monitorare in modo acustico se un circuito è collegato o meno, ad esempio nel caso di una macchina il cui utilizzo possa implicare pericolo, semplicemente collegando l'alimentazione il circuito emette un suono, che potremo selezionare in questo caso con il generatore di suoni, udendo in base alla frequenza scelta, da un semplice "clac, clac" fino a un suono acuto; quando invece si aziona il pulsante P1 inizia la temporizzazione del monostabile e il suono cambia nettamente.

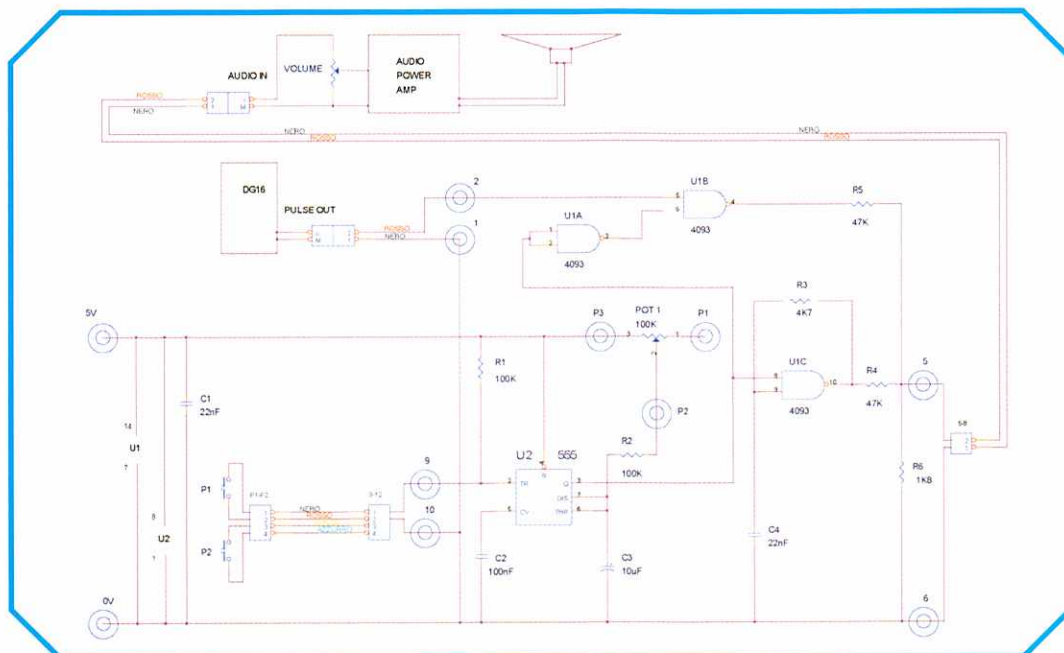
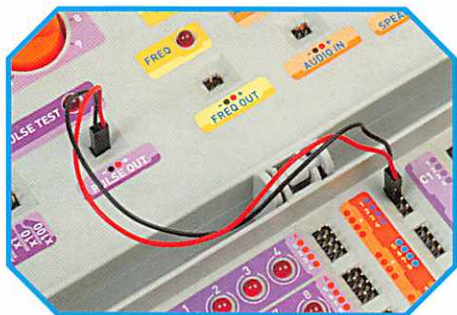
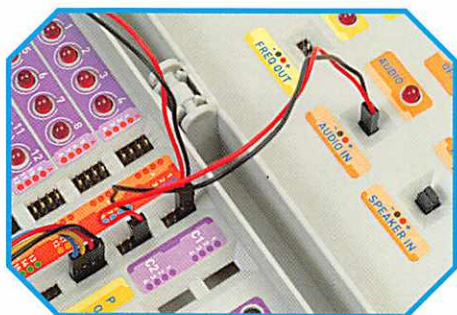
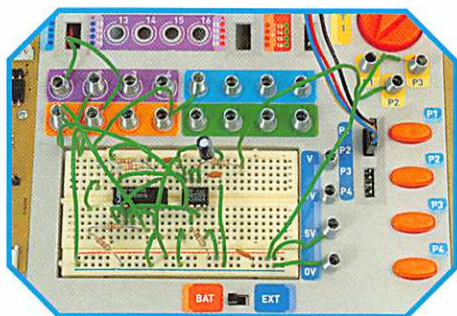
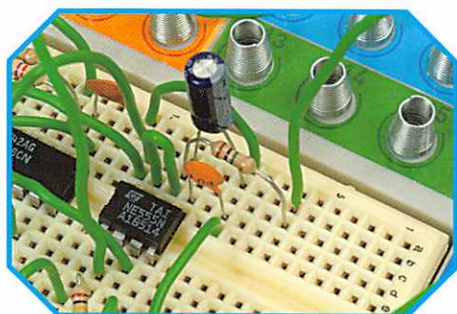
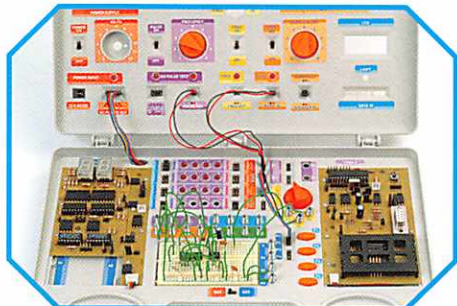
Il funzionamento del circuito si può capire facilmente spiegando ognuna delle sue parti.

Inizieremo dal monostabile che, come si può vedere dallo schema, è configurato su un circuito integrato 555 configurato da monostabile; il tempo in cui rimane operativo dopo l'attivazione, che si esegue con il pulsante P1, è determinato dalla capacità del condensatore C3 e dalla resistenza risultante dalla somma di R2 con il valore del potenziomentro.

La porta U1C si utilizza come oscillatore astabile per generare il segnale di allarme quando il monostabile è attivato, l'oscillatore si attiva ed emette il segnale quando il livello sul terminale 8 di U1 è alto, questo succede unicamente quando il monostabile è attivo.

L'oscillatore che genera il segnale di circuito acceso ma non attivo si prende dal generatore di toni del laboratorio e lo si porta all'in-

Componenti sulla scheda Bread Board.

Cablaggio della scheda.

Schema elettrico.

Collegamento a PULSE OUT.

Collegamento a AUDIO IN.

Vista generale del cablaggio.

Cambiando R2 varia la temporizzazione.

Vista dell'esperimento completato.

gresso del segnale audio tramite la porta U1B. Questa porta lascia passare il segnale quando l'altro ingresso, terminale 6, è a livello alto; come possiamo vedere, questo ingresso è collegato alla porta U1A configurata come porta invertente, quindi quando l'uscita del monostabile è a livello basso, questa porta inverte il livello generando un segnale alto sull'uscita, che si utilizza per far passare attraverso la porta U1B il segnale audio verso l'amplificatore.

Montaggio

Il montaggio si esegue come d'abitudine, inserendo in modo ordinato i componenti sulla scheda Bread Board, realizzando il cablaggio, e rivedendo tutto il lavoro prima di collegare il cavo di alimentazione alla molla da 5 V.

L'uscita del generatore di impulsi si prende da PULSE OUT, utilizzando un cavetto terminato su due connettori per portarla alle molle 1 e 2.

Il segnale audio di uscita si porta con un cavetto a due fili all'ingresso dell'amplificatore audio, AUDIO IN, utilizzando in questo caso le molle 5 e 6.

Il pulsante si collega con un cavetto a quattro fili, di cui ne utilizzeremo solamente due. È anche necessario inserire i due ponticelli in posizione orizzontale tra l'uscita AUDIO OUT e SPEAKER.

Il comando del volume deve essere vicino al minimo.

Prova

Dopo aver verificato tutto si collega l'alimentazione al laboratorio e si attivano i circuiti utilizzati con PULSE ON e AUDIO ON. Si deve poter ascoltare un suono sull'altoparlante, che si regola con FREQUENCY, premendo P1 il suono diventa continuo.

LISTA DEI COMPONENTI

U1	Circuito integrato 4093
U2	Circuito integrato 555
R1, R2	Resistenza 100 K (marrone, nero, giallo)
R3	Resistenza 4K7 (giallo, viola, rosso)
R4, R5	Resistenza 47 K (giallo, viola, arancio)
C1, C4	Condensatore 22 nF
C2	Condensatore 100 nF
C3	Condensatore 10 µF, elettrolitico

Generatore di frequenze: la pratica

Verifichiamo il funzionamento del nostro generatore di frequenze montando il suo circuito elettrico sul laboratorio. Eseguiremo due montaggi di uscita.
Nel primo, l'uscita sarà collegata a un diodo LED per poter vedere la risposta su quest'ultimo in base alle diverse frequenze.
Nel secondo, l'uscita sarà collegata all'amplificatore audio per ascoltare sull'altoparlante le frequenze generate.

Scrittura del programma sul microcontroller

Per eseguire la scrittura del PIC dobbiamo prima configurare il laboratorio. Caricheremo direttamente il programma sul PIC mediante IC-Prog, quindi dovremo configurare i ponticelli JP1, JP2 e JP3 della scheda DG06, JP8 e JP9 della scheda DG07. I primi tre dovranno essere inseriti sulle posizioni 1 e 2, mentre JP8 e JP9 devono avere i ponticelli inseriti. Collegheremo il cavo di comunicazione tra il PC e il laboratorio e apriremo IC-Prog.

Su IC-Prog selezioneremo il dispositivo PIC16F870 e apriremo il file "GENERA~1.HEX". Quando si compila e si assembla il file in assembler (.asm) se ne crea uno in codice macchina (.hex) con lo stesso nome dell'originale. Se il nome è più lungo di otto caratteri utilizza la compressione del nome che abbiamo visto (~1).

Configuriamo l'oscillatore come XT, la protezione del codice su CP OFF e attiveremo i bit della parola di configurazione WDT e PWRT.

Eseguiremo la scrittura del micro e verificheremo che il processo sia stato eseguito correttamente leggendone il contenuto.

Non dimenticate che prima di scrivere il PIC è necessario cancellarne il contenuto.

Montaggio

Ora prepareremo il laboratorio per montare il circuito elettrico che serve per provare il programma.

Scheda	Connettore	Scrittura	Funzionamento
DG06	JP1	Tra 1 y 2	Tra 2 y 3
	JP2	Tra 1 y 2	Tra 2 y 3
	JP3	Tra 1 y 2	Tra 2 y 3
DG07	JP8	Con ponticello	Senza ponticello
	JP9	Con ponticello	Senza ponticello

Configurazione dei ponticelli per i due modi di lavoro.

Configurazione per il modo lavoro

Prima di iniziare con il montaggio dobbiamo predisporre il laboratorio in modo lavoro. L'alimentazione la riceveremo dal laboratorio stesso, quindi i ponticelli di JP1, JP2 e JP3 li potremo inserire sulle posizioni 2 e 3. Libereremo i terminali RB6 e RB7 togliendo i ponticelli da JP8 e JP9. Infine toglieremo il cavo di comunicazione.

Montaggio degli ingressi

I terminali di ingresso con i quali selezioneremo le frequenze dell'onda di uscita sono RA2, RA1 e RA0. Sulla scheda Bread Board inseriremo tre resistenze che da un lato andranno al negativo (0 V) e dall'altro capo andranno alle molle di collegamento 9, 10 e 11. Dato che non abbiamo cavetti a sufficienza per eseguire il montaggio diretto dei pulsanti, realizzeremo un montaggio particolare per uno di essi inserendolo tra i terminali di ingresso e 5 V.

Uniremo con un filo i 5 V con la molla di collegamento 14 e con l'altro le molle 13 e 11.

Questo montaggio si può vedere nella figura. Con un cavetto a due fili uniamo il pulsante P3 con il connettore maschio corrispondente alle molle 13 e 14. Uniremo ora con un cavetto a quattro fili i terminali di ingresso RA2:RA0 con il connettore maschio corrispondente alle molle 9, 10 e 11.

Ci rimangono solamente da inserire gli altri due pulsanti tra il positivo e gli ingressi del micro. Con un cavetto eseguiremo il collegamento diretto dei pulsanti P1 e P2 alla scheda Bread Board.

Con questo montaggio abbiamo preparato tre ingressi, quindi ora passiamo a eseguire il montaggio relativo all'uscita.

Uscita su un diodo LED

Esistono diverse possibilità all'interno del laboratorio per provare che il nostro generatore di onda quadra multifrequenza funzioni correttamente.

Il più semplice è collegare all'uscita del PIC un diodo LED e vedere se riusciamo a percepire le variazioni nella velocità di lampeggio quando variamo la frequenza mediante i pulsanti di ingresso.

Uniremo, mediante un cavetto a quattro fili, i bit meno significativi della porta B con la matrice dei diodi LED e controlleremo che i ponticelli sui catodi siano inseriti.

La prova di funzionamento con il LED ci permette di verificare solamente l'esistenza di tre

IC-Prog 1.05C - Programmatore Prototipo - C:\Programmi\MPLAB\Progetti\GENERA~...

File Modifica Buffer Settaggi Comandi Strumenti Visualizza Aiuto

PIC 16F870

Address - Program Code

```
0000: 281B 3FFF 3FFF 3FFF 280E 0782 3400 3486
0008: 34C5 34DA 34E4 34EA 34EE 34F1 00A1 0E03
0010: 00A2 0820 0081 110B 3001 0686 0E22 0083
0018: 0EA1 0E21 0009 0186 1683 0186 3006 009F
0020: 301F 0085 3001 0081 1283 3020 008B 0064
0028: 0805 3907 1D03 282E 138B 2827 2005 00A0
0030: 178B 2827 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF
0038: 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF
0040: 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF
0048: 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF
0050: 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF
0058: 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF
```

Address - Eeprom Data

```
0000: FF FF FF FF FF FF FF FF   ÿÿÿÿÿÿÿÿ
0008: FF FF FF FF FF FF FF FF   ÿÿÿÿÿÿÿÿ
0010: FF FF FF FF FF FF FF FF   ÿÿÿÿÿÿÿÿ
0018: FF FF FF FF FF FF FF FF   ÿÿÿÿÿÿÿÿ
0020: FF FF FF FF FF FF FF FF   ÿÿÿÿÿÿÿÿ
0028: FF FF FF FF FF FF FF FF   ÿÿÿÿÿÿÿÿ
0030: FF FF FF FF FF FF FF FF   ÿÿÿÿÿÿÿÿ
0038: FF FF FF FF FF FF FF FF   ÿÿÿÿÿÿÿÿ
```

Configurazione

Oscillator: XT

Code Protect: CP OFF

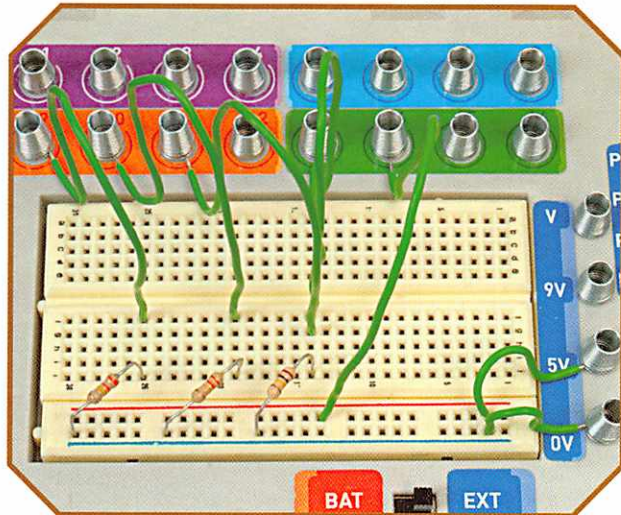
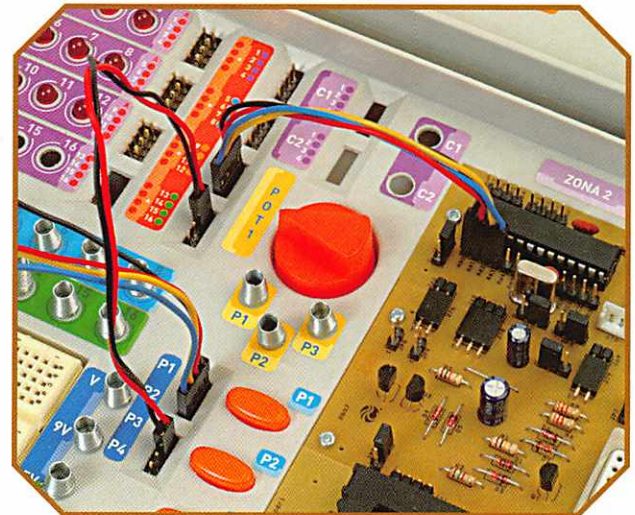
Fuses: WDT (✓), PWRT (✓), BODEN, LVP, CPD, WRT, DEBUGGER

Checksum: 0AFC — ID Value: FFFF

Config Word : 3D35h

Buffer 1 | Buffer 2 | Buffer 3 | Buffer 4 | Buffer 5

JDM Programmer su Com1 — Chip: PIC 16F870 (89)

Carichiamo su IC-Prog il programma del generatore.

Montaggio degli ingressi sulla scheda Bread Board.

Collegamenti dei cavetti di ingresso.

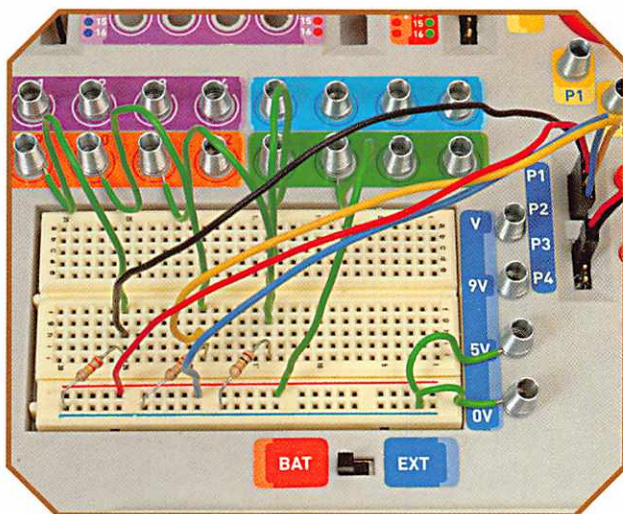
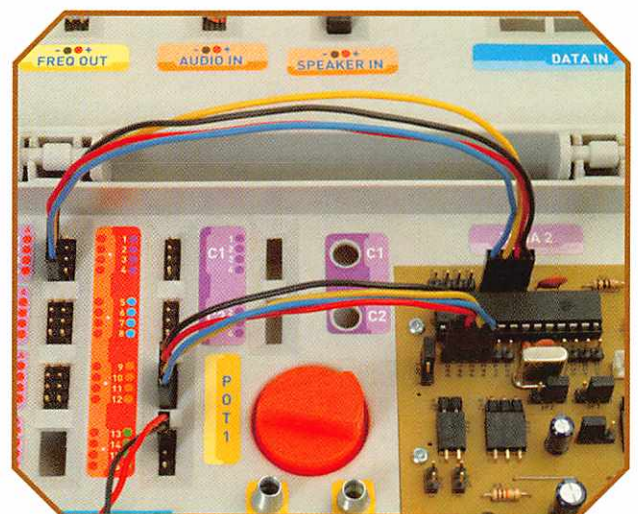
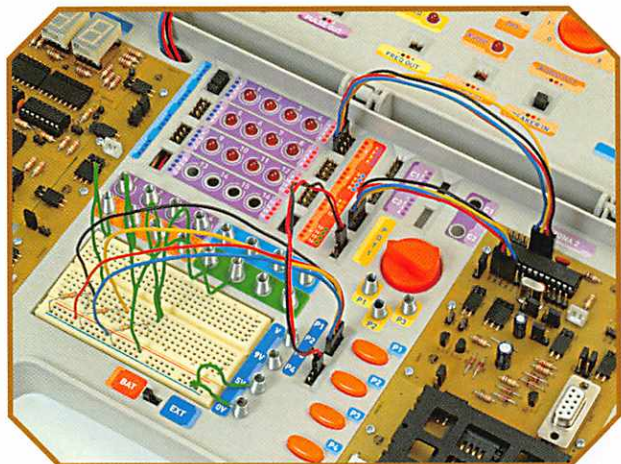
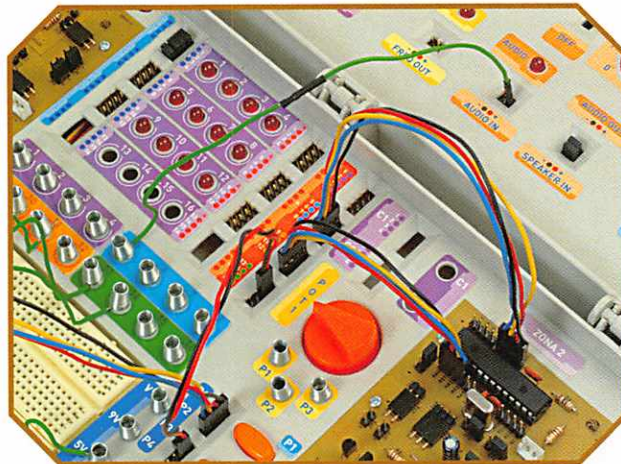
stati diversi: il LED rimarrà spento quando la frequenza è 0 e la porta si trova ferma a livello "basso", il LED si accenderà completamente quando la frequenza è 0 ma la porta si trova ferma a livello "alto" e il LED sarà in uno stato di "media accensione" quando si preme un tasto qualsiasi o diversi per volta.

In quest'ultimo caso verranno generate delle frequenze, ma la velocità con cui cambia lo stato del LED è talmente alta che l'occhio umano non è assolutamente in grado di distinguerla. Saremo in grado di percepire unicamente maggiore o minore luminosità sul LED.

Uscita sull'amplificatore audio

Proviamo ora un'altra delle alternative per poter osservare il funzionamento del nostro generatore.

Sfrutteremo l'amplificatore audio del laboratorio, collegheremo l'onda quadra generata all'ingresso di quest'ultimo e potremo così ascoltare come le diverse frequenze generano suoni diversi.

Scollegheremo il cavetto sulla matrice dei LED e lo collegheremo al connettore maschio corrispondente alle molle dalla 5 alla 8 (colo-

Inseriamo i due pulsanti rimanenti sul circuito.

Colleghiamo l'uscita al LED mediante un cavetto.

Montaggio completo con uscita su diodo LED.

Colleghiamo l'uscita del micro all'ingresso audio.

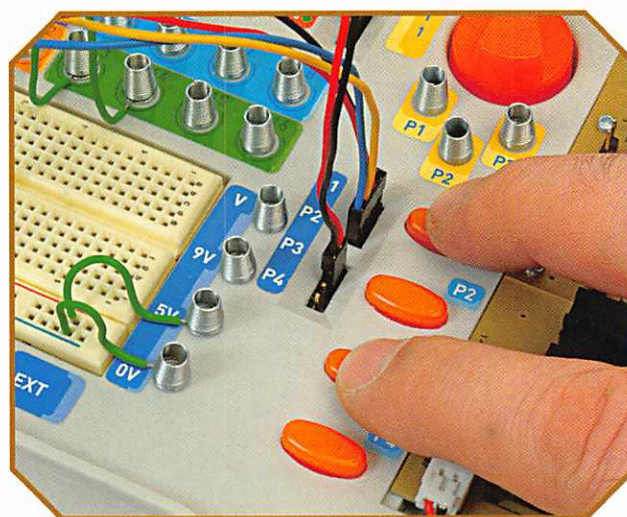
re azzurro). Alla molla di collegamento 5 arriva, pertanto, il segnale di uscita del micro, quindi collegheremo questa molla con un filo al connettore positivo (+) dell'ingresso audio (AUDIO IN). Nell'immagine si può vedere il montaggio descritto.

Il potenziometro della parte amplificatore audio serve per controllare il volume, quindi controlleremo che non sia a 0. Inseriremo i ponticelli che uniscono i segnali di uscita audio con l'ingresso dell'altoparlante e, infine, attiveremo l'interruttore AUDIO ON.

Se proviamo le diverse combinazioni di ingresso sentiremo che i suoni generati sono diversi, dato che cambia la frequenza dell'onda. Verificate anche cosa succede quando non ci sono pulsanti attivati.

Conclusioni

Mediante questo programma abbiamo interagito con un nuovo dispositivo esterno, ovvero con l'amplificatore audio. Esercitatevi con il programma del generatore, ampliando il range delle frequenze (inserendo più ingressi), modificando le frequenze di uscita, ecc. Potrete verificare che i suoni ottenuti saranno diversi, che il programma è stato ben progettato e risponde in modo soddisfacente alle richieste del progetto, inoltre vi servirà per acquisire confidenza nella programmazione.

Collegamento sull'ingresso dell'amplificatore audio.

Proviamo il funzionamento attivando i pulsanti.

Esercizio: il vostro turno, il programma

Se vi appassiona il mondo dell'elettronica e dei microcontroller applicati alla realtà, avrete sicuramente pensato in più di una occasione: potrei sviluppare questa applicazione con un microcontroller? Nella vita quotidiana possiamo trovare molti processi che possono essere automatizzati controllandoli con un PIC.

Enunciato

Vogliamo progettare una macchina di "Il Vostro Turno". In molte strutture commerciali di vendita al pubblico è necessario prendere un numero per essere serviti. Esiste un pannello visualizzatore dove il cliente può vedere il numero che è servito in quel momento e che il dipendente aggiorna mediante un pulsante.

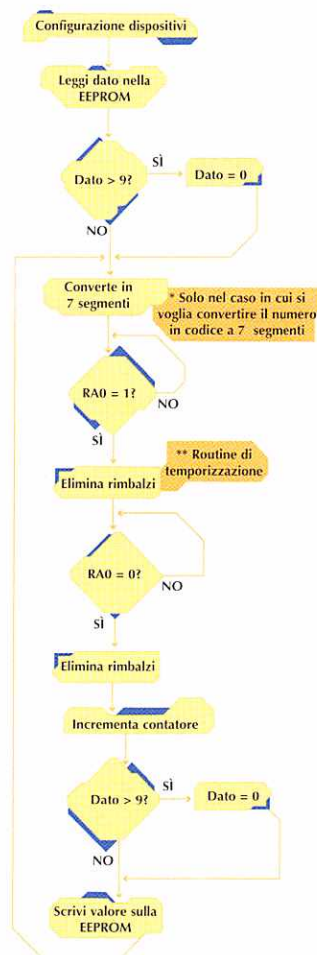
Per realizzare questa applicazione abbiamo bisogno di un pulsante di ingresso e di lavorare con i display a 7 segmenti del laboratorio. Inoltre vogliamo che, a fronte di un buco di tensione, il conteggio riprenda dall'ultimo numero visualizzato prima dell'evento. Questo significa che dovremo lavorare con la memoria EEPROM, dato che non è volatile.

Organigramma

Predisponiamo una soluzione generale all'applicazione. Siamo a conoscenza che la prima cosa da effettuare è la configurazione dei dispositivi con cui vogliamo lavorare (porte e temporizzatori), che per scrivere e leggere la EEPROM dovremo ricorrere alle subroutine già sviluppate (le copieremo da qualche codice già realizzato), e per collegare direttamente il display a 7 segmenti l'uscita deve essere fornita in questo codice per evitare rimbalzi che si possono generare attivando un pulsante, quindi è consigliabile utilizzare una routine di ritardo.

Analizziamo ora ciò che deve fare il programma. Inizialmente si deve leggere il valore scritto nella memoria EEPROM, dato che arriviamo da una mancanza di alimentazione. In base al valore inizieremo a contare da zero o da quest'ultimo valore. Fino a quando non si attiva il pulsante forniremo sull'uscita l'ultimo valore convertito in 7 segmenti, ma nel momento in cui RA0 vale 1, attenderemo il tempo di ritardo, dovuto al cambio di stato del segnale ed entreremo in un ciclo fino a quando il segnale non torna a 0. In questo momento, dopo il suo ritardo corrispondente, dedurremo che è stato generato un impulso e incrementeremo il contatore, visualizzando un nuovo numero e scrivendolo nella memoria EEPROM.

Dopo aver fatto questo torneremo ad attendere un altro cambio di stato su RA0, e ripeteremo la sequenza precedente.

Svilupperemo una macchina per "Il Vostro Turno".

Organigramma dell'applicazione.

```
;       ESERCIZIO: La memoria EEPROM dei dati. La macchina "IL VOSTRO TURNO"
;
;Si tratta di emulare il funzionamento delle macchine tipo "IL VOSTRO TURNO" comuni in molti
;negozi. Sul display LCD si visualizzerà il numero del turno attuale. Questo numero si incrementa
;ad ogni impulso applicato su RA0. Nella memoria EEPROM del PIC16F870 si scrive l'ultimo numero visualizzato,
;in questo modo, in caso di mancanza di tensione (ad esempio), si ricomincia il conteggio dall'ultimo numero.
;
;Si parte dal sistema che inizia per la prima volta, si visualiza lo 0

                List     p=16F870          ;Tipo di processore
                include  "P16F870.INC"     ;Definizioni dei registri interni

Contatore       equ      0x20              ;Variabile per il contatore

                org      0x00              ;Vector di Reset
                goto     Inizio
                org      0x05              ;Salva vector di interrupt

;*************************************************************************************
```

Intestazione del programma.

Codice

Siamo pronti per iniziare a progettare il nostro codice e inizieremo con l'intestazione del programma e i relativi commenti, le definizioni e le direttive di organizzazione nella memoria. Possiamo definire anche una variabile che funzioni come contatore.

Subroutine di scrittura e lettura nella EEPROM

In un progetto con i microcontroller difficilmente vi è qualcosa di più prezioso rispetto al tempo di progettazione.

Il tempo è denaro e per questo dobbiamo risparmiarne il più possibile quando sviluppiamo un progetto. Nel primo esercizio in cui abbiamo lavorato con la memoria EEPROM abbiamo realizzato una subroutine di scrittura e

```
;*************************************************************************************
;EE_Write: Scrive un byte nella EEPROM dei dati. L'indirizzo sarà contenuto in EEADR e
;si suppone che il dato sia stato precedentemente caricato su EEDATA

EE_Write        bsf      STATUS,RP1
                bsf      STATUS,RP0        ;Passiamo al banco 3
                bcf      EECON1,EEPGD      ;Selezioniamo la EEPROM dei dati
                bsf      EECON1,WREN       ;Abilitiamo la sua scrittura
                movlw    0x55
                movwf    EECON2
                movlw    0xaa
                movwf    EECON2            ;Sequenza obbligatoria
                bsf      EECON1,WR         ;Iniziamo la scrittura
                bcf      STATUS,RP0
                bcf      STATUS,RP1        ;Torniamo al banco 0
ATTENDI         btfss    PIR2,EEIF         ;Attendiamo che termini la scrittura
                goto     ATTENDI
                bcf      PIR2,EEIF         ;Resettiamo il flag della EEPROM
                return

;*************************************************************************************
;EE_Read: Legge un byte della EEPROM. Si suppone che il registro EEADR sia stato caricato
;con l'indirizzo da leggere. Su EEDATA apparirà il dato letto.

EE_Read         bsf      STATUS,RP0
                bsf      STATUS,RP1        ;Passiamo al banco 3
                bcf      EECON1,EEPGD      ;Selezioniamo la EEPROM dei dati
                bsf      EECON1,RD         ;Abilitiamo la sua la lettura
                bcf      STATUS,RP1
                bcf      STATUS,RP0        ;Selezione del banco 0
                return

;*************************************************************************************
```

Subroutine di scrittura e lettura della EEPROM.

una di lettura. Ogni volta che lavoreremo con la memoria EEPROM potremo utilizzare queste subroutine, anche se in qualche caso dovremo fare qualche leggero ritocco. In questo modo guadagnamo tempo ed evitiamo possi-

```
;*************************************************************************************
;
;Tabella: Questa routine converte il codice BCD presente sui bit meno significativi
;del reg. W nel loro equivalente a 7 segmenti. Il codice a 7 segmenti rimane anche
;sul reg. W

Tabella:        addwf    PCL,F             ;Spostamento sulla tabella
                retlw    b'00111111'       ;Digit 0
                retlw    b'00000110'       ;Digit 1
                retlw    b'01011011'       ;Digit 2
                retlw    b'01001111'       ;Digit 3
                retlw    b'01100110'       ;Digit 4
                retlw    b'01101101'       ;Digit 5
                retlw    b'01111101'       ;Digit 6
                retlw    b'00000111'       ;Digit 7
                retlw    b'01111111'       ;Digit 8
                retlw    b'01100111'       ;Digit 9

;*************************************************************************************
```

Subroutine di conversione in codice a 7 segmenti.

vsturno - Blocco note

File Modifica Formato Visualizza ?

```
;**************************************************************************************
;Delay_20_ms: Questa routine di temporizzazione ha come obiettivo l'eliminazione
;dell'effetto rimbalzo dei dispositivi elettromeccanici. Realizza un ritardo di 20 ms.
;Se il PIC lavora ad una frequenza di 4MHz, il TMR0 evolve ogni µs. Se vogliamo tempo-
;rizzare 20000 µs (20 ms) con un prescaler de 128, il TMR0 dovrà contare 156 eventi
;(156 * 128). Il valore 156 equivale a 9C hex. e dato che il TMR0 è ascendente lo
;dovremo caricare con il suo complemento a 1 (63 hex.).

Delay_20_ms:    bcf     INTCON,T0IF     ;Azzera il flag di overflow
                movlw   0x63            ;Complemento hex. di 156
                movwf   TMR0            ;Carica il TMR0
Delay_20_ms_1   clrwdt                  ;Aggiorna il WDT
                btfss   INTCON,T0IF     ;Overflow del TMR0?
                goto    Delay_20_ms_1   ;Non ancora
                bcf     INTCON,T0IF     ;Ora sì, azzera il flag
                return
```

Subroutine di ritardo anti-rimbalzo.

vsturno - Blocco note

File Modifica Formato Visualizza ?

```
Inizio          clrf    PORTB           ;Cancella i latch di uscita
                bsf     STATUS,RP0      ;Seleziona banco 1
                clrf    TRISB           ;Porta B si configura come uscita
                movlw   b'00000110'
                movwf   ADCON1          ;Configuriamo la porta come I/O digitali
                movlw   b'00011111'
                movwf   TRISA           ;RA0-RA4 ingressi RA5-RA7 uscite
                movlw   b'00000110'
                movwf   OPTION_REG      ;Prescaler di 128 per il TMR0
                bsf     STATUS,RP1      ;Seleziona banco 2
```

Configurazione dei dispositivi.

bili complicazioni. Nella figura possiamo vedere le due subroutine che verranno incluse nel programma.

Subroutine di conversione in codice a 7 segmenti

Come abbiamo visto per la memoria EEPROM, per convertire un numero in codice a 7 segmenti possiamo utilizzare nel nostro codice una subroutine che abbiamo realizzato nell'esercizio precedente. Questa parte è opzionale, dato che nell'enunciato non si specifica il formato dell'uscita, quindi potremo fornire l'uscita in codice a 7 segmenti oppure in binario, e far convertire il dato ai driver del display a 7 segmenti. Nel caso in cui si volesse realizzare quest'ultima opzione, sarebbe sufficiente inserire ";" su ogni linea della subroutine e a ogni chiamata a questa con l'istruzione "call Tabella". Impostando il segno "punto e virgola" il compilatore considererà queste linee come commenti.

Subroutine di ritardo anti-rimbalzo

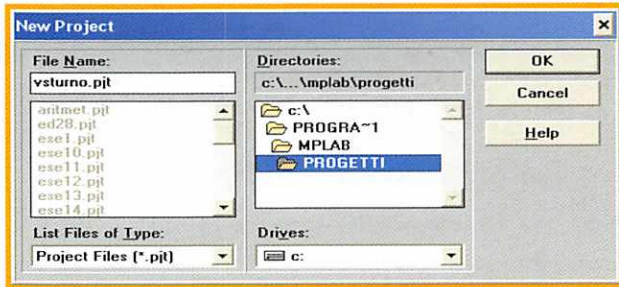
Quando attiviamo un elemento elettromeccanico possiamo produrre l'effetto rimbalzo. Questo fa sì che in un tempo molto breve il segnale possa cambiare di stato fino a quando si stabilizza. Il PIC può rilevare queste fluttuazioni e interpretarle in modo sbagliato. Dobbiamo inserire una subroutine che generi un ritardo ogni volta che c'è un cambio di stato sul segnale di ingresso, come per le subroutine precedenti, anche questa è già stata utilizzata in altri casi. Copieremo la subroutine e la inseriremo nel nostro programma.

Programma principale

Il programma principale deve iniziare configurando i dispositivi con cui vogliamo lavorare. Sappiamo che la porta B sarà la porta di uscita e la A di ingresso, per i segnali digitali, e che dobbiamo configurare il registro OPTION_REG per lavorare con il TMR0 che utiliz-

```
vsturno - Blocco note
File  Modifica  Formato  Visualizza  ?

            clrf    EEADR           ;Seleziona indirizzo 00 della EEPROM
            call    EE_Read         ;Leggi byte della EEPROM
            bsf     STATUS,RP1      ;Seleziona banco 2
            movlw   0x09
            subwf   EEDATA,W
            bcf     STATUS,RP1      ;Seleziona banco 0
            btfsc   STATUS,C        ;Maggiore di 9?
            goto    Ini_0           ;Si, imposta a 0 il contatore
            goto    Ini_1           ;No
Ini_0       clrf    Contatore       ;Imposta a 0 il contatore
            goto    Loop
Ini_1       bsf     STATUS,RP1      ;Seleziona banco 2
            movf    EEDATA,W
            bcf     STATUS,RP1      ;Seleziona banco 0
            movwf   Contatore       ;Inizializza il contatore

Loop        movf    Contatore,W
            call    Tabella         ;Converte il contatore a 7 segmenti
            movwf   PORTB           ;Visualizza sul display

Wait_0      clrwdt                  ;Aggiorna il WDT
            btfss   PORTA,0         ;RA0 è a "1"?
            goto    Wait_0          ;No, attendere
            call    Delay_20_ms     ;Eliminare rimbalzi

Wait_1      clrwdt                  ;Aggiorna il WDT
            btfsc   PORTA,0         ;RA0 è a "0"?
            goto    Wait_1          ;No, attendere
            call    Delay_20_ms     ;Eliminare rimbalzi. C'è stato un impulso

            incf    Contatore,F     ;Incrementa contatore
            movlw   .10
            subwf   Contatore,W
            btfsc   STATUS,Z        ;Contatore maggiore di 9?
            clrf    Contatore       ;Si, torna a 00
            movf    Contatore,W
            bsf     STATUS,RP1      ;Seleziona banco 2
            movwf   EEDATA
            call    EE_Write        ;Scrive il nuovo valore del contatore nella EEPROM
            goto    Loop

            end                     ;Fine del programma sorgente
```

Codice che risolve l'applicazione.

za la subroutine di ritardo. Con i dispositivi configurati seguiremo il flusso dell'organigramma codificandolo in istruzioni. Leggeremo la EEPROM e osserveremo il valore che contiene. Visualizziamo e attendiamo che si attivi l'ingresso. In quel momento dovremo eseguire il ciclo necessario per considerare valido l'impulso (ritardo, attesa di ritorno a 0, ritardo). Successivamente incrementeremo il contatore verificando che il suo valore sia minore di 9, dato che il valore del conteggio successivo è 0. Infine passeremo il valore alla EEPROM e torneremo a ripetere il ciclo dal punto in cui è stata chiamata la tabella portando il valore sull'uscita.

Creiamo un progetto su MPLAB.

Compilazione

Come avete potuto verificare abbiamo risolto un'applicazione complessa sviluppando un codice semplice e inserendo delle subroutine che avevamo già sviluppato per esercizi precedenti. Ora dobbiamo verificare di non aver commesso errori sia nell'includere delle subroutine che nell'indirizzarci a esse e anche nelle strutture sintattiche delle istruzioni.

Apriamo MPLAB, creiamo un nuovo progetto e alleghiamo a esso il nostro codice.

Fatto questo, apriamo il file con il codice per visualizzarlo sul monitor e correggerlo nel caso fosse necessario, selezioneremo poi l'opzione Build All. Il codice si compila senza errori anche se il compilatore ci informa circa 15 possibili errori di indirizzamento dei registri nella memoria. Nell'immagine possiamo osservare il risultato ottenuto dalla compilazione del programma.

```
Build Results
Building VSTURNO.HEX...

Compiling VSTURNO.ASM:
Command line: "C:\PROGRA~1\MPLAB\MPASMWIN.EXE /p16F870 /q C:\PROGRA~1\MPLAB\PROGETTI\VSTURNO.ASM"
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\VSTURNO.ASM 25 : Register in operand not in bank 0.  Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\VSTURNO.ASM 26 : Register in operand not in bank 0.  Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\VSTURNO.ASM 28 : Register in operand not in bank 0.  Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\VSTURNO.ASM 30 : Register in operand not in bank 0.  Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\VSTURNO.ASM 31 : Register in operand not in bank 0.  Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\VSTURNO.ASM 45 : Register in operand not in bank 0.  Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\VSTURNO.ASM 46 : Register in operand not in bank 0.  Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\VSTURNO.ASM 89 : Register in operand not in bank 0.  Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\VSTURNO.ASM 91 : Register in operand not in bank 0.  Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\VSTURNO.ASM 93 : Register in operand not in bank 0.  Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\VSTURNO.ASM 95 : Register in operand not in bank 0.  Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\VSTURNO.ASM 98 : Register in operand not in bank 0.  Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\VSTURNO.ASM 102 : Register in operand not in bank 0.  Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\VSTURNO.ASM 110 : Register in operand not in bank 0.  Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\VSTURNO.ASM 135 : Register in operand not in bank 0.  Ensure

Build completed successfully.
```

Risultato della compilazione.